

## REMARKS

### Rejection of Claims on Art Grounds in the 06/17/2004 Office Action, and Traversal Thereof

In the 06/17/2004 Office Action, claims 1-40 were rejected on prior art grounds, under 35 U.S.C. 102(e) and 35 U.S.C 103(a). Claims 1, 3-8, 12-27, and 31-40 were rejected under 35 U.S.C. 102(e) as being anticipated by Underwood (U.S. Pat. 6,523,027 B1). Claims 2, 9-11, and 28-30, were rejected under 35 U.S.C. 103(a) as being unpatentable over Underwood. Applicant asserts that the Underwood patent (U.S. Pat. 6,523,027 B1) cited and relied upon by the examiner does not anticipate nor does it make obvious the claims of the current invention. The above rejections of the claims 1-40 on the stated art grounds are traversed, and consideration of the patentability of the claims 1-40 is requested, in light of the following remarks.

### Arguments for Patentability

The Underwood patent describes an improved method and system for designing software solutions to eCommerce needs by facilitating the communication of *business information between computer servers* to fulfill eCommerce needs. The Underwood invention overcomes many of the previous limitations of large custom-built solutions by incorporating object-oriented programming and framework design common to the development of modern software applications. In so doing, the software project designs are far simpler to create and less redundant, the resulting functions integrate well with third party applications, and there is greater ease in updating the applications to advancements in the software industry by simply altering the existing frameworks and directories.

By contrast, the present invention relates to a software development tool capable of simultaneously displaying graphical and textual views of source code using a language-neutral representation of the source code to generate the two views. The software development tool stores the language-neutral representation in a transient meta-model to allow changes made to either to the graphical or textual views to be immediately and automatically committed back to the source code, and vice versa, for any given programmer. This is distinguishable over prior art software development tools that only saved the working copy of the software project in a repository, thus requiring the user to separately commit the changes to the source code file. This "batch" style method has a number of disadvantages including: redundancy between the repository and source code files, wasted processing time to rewrite the entire source code to update, occasional overwriting of desired changes to the source code, and limitations to a single programming language.

On the other hand, the present invention provides "round trip engineering" to incrementally commit the changes to either the textual or graphical views and apply them to the source code, and vice versa. This novel method of persistent updating source code and graphical/textual views abolishes the need for a repository and allows the modifications made to the software project to be committed immediately and automatically. In addition, the present invention further provides a version control system to allow different programmers to separately create and modify their own versions of a given software project. The version control system operates in part to automatically maintain and track the history of incremental changes made to a particular version of the software program and stores this history in a specified central repository. Furthermore,

the version control system is in-built to operate *seamlessly* in association with the software development tool, thus facilitating the ability of each programmer to perform a variety of version-control functions with their own working copy stored on the individual programmer's computer (client component) in relation with the master copy stored in the central repository of a different computer (server component). By contrast, the version control systems in Underwood such as Visual SourceSafe are operated in *separate windows* from the software development tool in contrast to the present invention which unifies the version control system and the novel software development tool into a single graphical interface.

The examiner has indicated that prior art software development tools and languages (such as UML referenced in Underwood and Fowler) have provided simultaneous graphical and textual representations of the source code in the context of software development for some time, but *these prior art methods and systems required the use of a repository to store the software project and posed numerous disadvantages as outlined above*. None of the prior art references including Underwood describe a type of "round trip engineering" in software design as disclosed by the present invention and particularly contrasted by figures 1 and 2 of the present patent application, allowing the simultaneous and automatic updating of the source code as well as the textual and graphical views. In addition, the present invention allows the retrieval and development of source code in a variety of program languages and conversion into a language-neutral representation, whereas the invention described in the Underwood patent is specifically targeted to developing *Java* (and closely related *Active X*) based web programs for B2B

eCommerce solutions to provide communication between servers and third party applications.

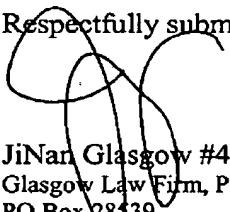
It is further conceded that prior art software development tools have utilized version control systems to manage the checking in and out of different software project versions by separate project developers. For example, such a device is included by reference with the Underwood invention citing the use of either Visual SourceSafe/Rational Rose or PVCS Version Manager depending on the type of application being developed. However, neither of these applications in the context of the Underwood invention (or any other to date) prescribes the use of a version control system *in concert with* a software development tool that provides simultaneous viewing of a diagrammatic and textual representation of the project source code wherein the traditional use of a repository to store the working copy of the software project is abolished and changes made to either display or to the source code itself are automatically, immediately, and incrementally reflected in the other view as well as the source code file.

## CONCLUSION

In view of the foregoing, claims 1-44, constituting the claims pending in the application, are submitted to be fully patentable and in allowable condition to address and overcome the rejections. If any issues remain outstanding, incident to the allowance of the application, Examiner Curcio is respectfully requested to contact the undersigned attorney at (919) 664-8222 or via email at [jnang@trianglepatents.com](mailto:jnang@trianglepatents.com) to discuss the resolution of such issues, in order that prosecution of the application may be concluded favorably to the applicant, consistent with the applicant's making of a substantial advance

in the art and particularly pointing out and distinctly claiming the subject matter that the applicant regards as the invention. This office action response is being submitted via fax to the USPTO on 09/16/2004.

Respectfully submitted,



JiNan Glasgow #42585  
Glasgow Law Firm, PLLC  
PO Box 28539  
Raleigh, NC 27611-8539  
919-664-8222  
919-664-8625 (fax)